From Non-Adaptive to Adaptive Pseudorandom Functions

Itay Berman Iftach Haitner*

September 15, 2011

Abstract

Unlike the standard notion of pseudorandom functions (PRF), a *non-adaptive* PRF is only required to be indistinguishable from random in the eyes of a *non-adaptive* distinguisher (i.e., one that prepares its oracle calls in advance). A recent line of research has studied the possibility of a *direct* construction of adaptive PRFs from non-adaptive ones, where direct means that the constructed adaptive PRF uses only few (ideally, constant number of) calls to the underlying non-adaptive PRF. Unfortunately, this study has only yielded negative results, showing that "natural" such constructions are unlikely to exist (e.g., Myers [EUROCRYPT '04], Pietrzak [CRYPTO '05, EUROCRYPT '06]).

We give an affirmative answer to the above question, presenting a direct construction of adaptive PRFs from non-adaptive ones. Our construction is extremely simple, a composition of the non-adaptive PRF with an appropriate pairwise independent hash function. In particular, the resulting PRF only makes a *single* call to the non-adaptive PRF.

1 Introduction

A pseudorandom function family (PRF), as defined by Goldreich, Goldwasser, and Micali [9], is a function family that cannot be distinguished from a family of truly random functions by any PPT (probabilistic polynomial-time) distinguisher that accesses a random member of the family as an oracle. PRFs have an extremely important role in cryptography, allowing parties that share a common key, to send secure messages, identity themselves and to authenticate messages [8, 11]. In addition, they have many other applications, essentially in any setting that requires random function provided as black-box [1, 2, 5, 6, 12, 16].

Different PRF constructions are known in the literature, whose security is based on different hardness assumption. Constructions relevant to this work are those based on the existence of pseudorandom generators [9] (and thus on the existence of one-way functions [10]), and on, the so called, synthesizers [15]. In this work we study the question of constructing (adaptive) PRFs from *non-adaptive* PRFs. The latter primitive is a (weaker) variant of the standard PRF we mentioned above, whose security is only guaranteed to hold against non-adaptive distinguishers (i.e., ones that "write" all their queries before the first oracle call). Since a non-adaptive PRF can be easily cast as a pseudorandom generator or as a synthesizer, [9, 15] tell us how to construct (adaptive) PRF for a non-adaptive one. In both of these constructions, however, the resulting (adaptive) PRF makes $\Theta(n)$ calls to the underlying non-adaptive PRF (where n being the input length of the

*School of Computer Science, Tel Aviv University. E-mail: `iftachh@cs.tau.ac.il, itayberm@post.tau.ac.il`.

functions).[1] A recent line of work has tried to figure out whether this number of calls is necessary. In a sequence of work [14, 17, 18, 4], it was shown that standard approaches (e.g., composition or XORing members of the non-adaptive family with itself) are unlikely to work. See more in Section 1.3.

1.1 Our Result

We show that a simple composition a non-adaptive PRF with an appropriate pairwise independent hash function, yields an adaptive PRF of similar security.

To state our result more formally, we use the following definitions: a function family $\mathcal{F}$ is $T = T(n)$-adaptive PRF, if no distinguisher of running time at most T, can tell a random member of $\mathcal{F}$ from random with advantage larger than $1/T$. Where if $\mathcal{F}$ is T-*non*-adaptive PRF, the above is only guarantee to holds against non-adaptive distinguishers. Given two function families $\mathcal{F}_1$ and $\mathcal{F}_2$, we let $\mathcal{F}_1 \circ \mathcal{F}_2$ [resp., $\mathcal{F}_1 \bigoplus \mathcal{F}_2$] be the function family whose members are all pairs $(f, g) \in \mathcal{F}_1 \times \mathcal{F}_2$, and the action $(f, g)(x)$ is defined as $f(g(x))$ [resp., $f(x) \oplus g(x)$].

We have the following results (see Section 3 for the formal statements).

Theorem 1.1 (Informal)**.** *Let $\mathcal{F}$ be a $p(n)T(n)$-non-adaptive PRF, where $p \in \mathrm{poly}$ is universal, and let $\mathcal{H}$ be an efficient pairwise-independent function family mapping strings of length n to $[T(n)]_{\{0,1\}^n}$, where $[T]_{\{0,1\}^n}$ is the first T elements (in lexicographic order) of $\{0,1\}^n$. Then $\mathcal{F} \circ \mathcal{H}$ is a $\sqrt[3]{T(n)}/2$-adaptive PRF.*

The above theorem is useful whenever T is polynomial-time computable (in this case, the family $\mathcal{H}$, assumed by the theorem, exists). For other cases, we have the following result.

Corollary 1.2 (Informal)**.** *Let $\mathcal{F}$ be a $T(n)$-non-adaptive PRF, let $\mathcal{H} = \{\mathcal{H}_n\}_{i\in\mathbb{N}}$ be an efficient length-preserving pairwise-independent family, and for $n \in \mathbb{N}$ and $i \leq \log n$, let $\mathcal{H}_n^i$ be the following "truncation" of $\mathcal{H}_n$: $\mathcal{H}_n^i = \{\widehat{h} \colon h \in \mathcal{H}_n\}$, where $\widehat{h}(x)$ is the $h(x)_{1,\ldots,\log(m_n(i))}$'th element in $\{0,1\}^n$, for $m_n(i) = n^{2^i \cdot \log n}$.*

Then for any polynomially-computable integer function $k(n) \leq \mathrm{poly}(n)$, the ensemble $\{\bigoplus_{i\in[k(n)]}(\mathcal{F}_n \circ \mathcal{H}_n^i)\}_{n\in\mathbb{N}}$ is a $\sqrt[3]{m_n(i)}/2$-adaptive PRF, for any polynomially-computable function $i(n) \leq k(n)$ with $m_n(i) \leq T(n)/q(n)$, where $q \in \mathrm{poly}$ is universal.

In particular, by applying Corollary 1.2 to a super-polynomial non-adaptive PRF $\mathcal{F}$ is (i.e., $\mathcal{F}$ is indistinguishable from random by any non-adaptive polynomial-time distinguisher) and a polynomial-time computable $k(n) \in \omega(1)$ (e.g., $k(n) = \log^*(n)$), we get a super-polynomial (adaptive) PRF, that makes $k(n)$ calls to $\mathcal{F}$.

Finally, we mentioned that both results are proven in a black-box manner, making the reductions fully-black-box.[2]

1.2 Proof Idea

The proof of Theorem 1.1 is carried through the following steps: first we show that $\mathcal{F} \circ \mathcal{H}$ is indistinguishable from $\Pi \circ \mathcal{H}$, where Π be the set of *all* functions from $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$ (letting

[1] In case one is only interested in super-polynomial adaptive PRF, then $w(\log n)$ calls are sufficient (see [7, Exe. 30]).

[2] In such a fully-black-box reduction, the adaptive-PRF construction and its proof of security, access the non-adaptive PRF and a potential adversary, as oracles.

$\ell(n)$ be $\mathcal{F}$'s output length), and then conclude the proof showing that $\Pi \circ \mathcal{H}$ is indistinguishable from Π.

$\mathcal{F} \circ \mathcal{H}$ **is indistinguishable from** $\Pi \circ \mathcal{H}$**.** Let D be (a possibly adaptive) algorithm of running-time $T(n)$, which distinguishes $\mathcal{F} \circ \mathcal{H}$ from $\Pi \circ \mathcal{H}$ with advantage $\varepsilon(n)$. We show how to use D to build a *non-adaptive* distinguisher $\widehat{\mathsf{D}}$ of running-time $p(n)T(n)$, which distinguishes $\mathcal{F}$ from Π with advantage $\varepsilon(n)$. Given oracle access to a function ϕ, the distinguisher $\widehat{\mathsf{D}}^{\phi}(1^n)$ first queries ϕ on the elements of $[T]_{\{0,1\}^n}$. Then it chooses at uniform $h \in \mathcal{H}$, and uses the answers to its T queries to emulate $\mathsf{D}^{\phi \circ h}(1^n)$.

Note that $\widehat{\mathsf{D}}$ can be implemented in time $p(n) \cdot T(n)$, for large enough $p \in \text{poly}$, and distinguishes $\mathcal{F}$ from Π with advantage $\varepsilon(n)$. Since $\widehat{\mathsf{D}}$'s queries to ϕ are non adaptive, the assumed security of $\mathcal{F}$ yields that $\varepsilon(n) < 1/p(n)T(n)$.

$\Pi \circ \mathcal{H}$ **is indistinguishable from** Π**.** We prove that $\Pi \circ \mathcal{H}$ is *statistically* indistinguishable from Π. Namely, even an unbounded distinguisher (that makes bounded number of calls) cannot distinguish between the families. The idea of the proof is fairly simple. Let D be an s-query algorithm trying to distinguish between $\Pi \circ \mathcal{H}$ and Π. We first note that the distinguishing advantage of D is bounded by its probability of finding a collision in a random $\phi \in \Pi \circ \mathcal{H}$ (in case no collision occurs, ϕ's output is uniform). We next argue that in order to find a collision in $\phi = (f, h) \in \Pi \circ \mathcal{H}$, the distinguisher D gains nothing from being adaptive. Indeed, assuming that D found no collision until the i'th call, then it has only learned that h does not collide on these first i queries. Therefore, a random (or even a constant) query as the $(i+1)$ call, has the same chance to yield a collision, as any other query has. Hence, we assume without loss of generality that D is non-adaptive, and use the pairwise independence of $\mathcal{H}$ to conclude that D's probability in finding a collision, and thus its distinguishing advantage, is bounded by $2s(n)^2/(T(n) - 2s(n)^2)$.

Combining the above, we get that no (adaptive) distinguisher whose running time is bounded by $\frac{1}{2}\sqrt[3]{T(n)}$, distinguishes $\mathcal{F} \circ \mathcal{H}$ from Π (i.e., from a random function) with advantage better than $\frac{T(n)^{\frac{2}{3}}/2}{T(n)-T(n)^{\frac{2}{3}}/2} + \frac{1}{p(n)T(n)} \leq 2/\sqrt[3]{T(n)}$. Namely, $\mathcal{F} \circ \mathcal{H}$ is $\sqrt[3]{T(n)}/2$ (adaptive) PRF.

1.3 Related Work

Maurer and Pietrzak [13] where he first to consider the question of building adaptive PRF from non-adaptive ones. They show that in the *information theoretic* model, composition of two non-adaptive PRFs *does* yield an adaptive one. (specifically, a T-non-adaptive PRF yields a $T(1+\ln\frac{1}{T})$-adaptive-PRF).

In contrast to the above, Myers [14] proved that in the computation model (that we consider here), it is impossible to reprove the result of [13] via fully-black-box reductions. Pietrzak [17] showed that if the Decisional Diffie-Hellman (DDH) assumption holds, then composition does not imply adaptive security, where in [18] he showed that if the composition of two non-adaptive PRF's is not adaptively secure, then a key-agreement protocol exists. Finally, Cho et al. [4] generalized [18] by proving that composition of two non-adaptive PRFs is not adaptively secure, iff (uniform transcript) key agreement protocol exists. We mention that [14, 17, 4], and in a sense also [13], hold also with respect to XORing of the non-adaptive families.

2 Preliminaries

2.1 Notations

We use calligraphic letters to denote sets, uppercase for random variables, and lowercase for values. For integer t, we let $[t] = \{1, \ldots, t\}$, and for a set $\mathcal{S} \subseteq \{0,1\}^*$ with $|\mathcal{S}| \geq t$, we let $[t]_{\mathcal{S}}$ the first t elements (in increasing lexicographic order) of $\mathcal{S}$. A function $\mu \colon \mathbb{N} \to [0,1]$ is *negligible*, if $\mu(n) = n^{-\omega(1)}$. We let poly denote the set all polynomials, and let PPT denote the set of probabilistic algorithms (i.e., Turing machines) that run in *strict* polynomial time.

Given a random variable X, we write $X(x)$ to denote $\Pr[X = x]$, and write $x \leftarrow X$ to indicate that x is selected according to X. Similarly, given a finite set $\mathcal{S}$, we let $s \leftarrow \mathcal{S}$ denote that s is selected according to the uniform distribution on $\mathcal{S}$.

The *statistical distance* of two distributions P and Q over a finite set $\mathcal{U}$, denoted as $\mathrm{SD}(P,Q)$, is defined as $\max_{\mathcal{S} \subseteq \mathcal{U}} |P(\mathcal{S}) - Q(\mathcal{S})| = \frac{1}{2} \sum_{u \in \mathcal{U}} |P(u) - Q(u)|$.

2.2 Function Families

Let $\mathcal{F} = \{\mathcal{F}_n \colon \mathsf{D}_n \mapsto \mathcal{R}_n\}_{n \in \mathbb{N}}$ stands for an ensemble of function families, where each $f \in \mathcal{F}_n$ has domain D_n and its range contained in $\mathcal{R}_n \subseteq \{0,1\}^{\ell(n)}$. For a function $\ell = \ell(n) \in \mathbb{N}$, we let $\Pi_\ell = \{\Pi_{n,\ell(n)}\}_{n \in \mathbb{N}}$, where $\Pi_{n,\ell(n)}$ is the set of all functions from $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$.

Definition 2.1 (efficient function family). *An ensemble of function families $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$ is* efficient *(for short, $\mathcal{F}$ is an efficient function family), if the following hold:*

Samplable. *$\mathcal{F}$ is samplable in polynomial-time: there exists a* PPT *that given 1^n, outputs (the description of) a uniform element in $\mathcal{F}_n$.*

Efficient. *There exists a polynomial-time algorithm that given $x \in \{0,1\}^n$ and (a description of) $f \in \mathcal{F}_n$, outputs $f(x)$.*

2.2.1 Operating on Function Families

Definition 2.2 (composition of function families). *Let $\mathcal{F}^1 = \{\mathcal{F}^1_n \colon \mathsf{D}^1_n \mapsto \mathcal{R}^1_n\}_{n \in \mathbb{N}}$ and $\mathcal{F}^2 = \{\mathcal{F}^2_n \colon \mathsf{D}^2_n \mapsto \mathcal{R}^2_n\}_{n \in \mathbb{N}}$ be two ensembles of function families with $\mathcal{R}^1_n \subseteq \mathsf{D}^2_n$ for every n. We define the* composition *of $\mathcal{F}^1$ with $\mathcal{F}^2$ as $\mathcal{F}^2 \circ \mathcal{F}^1 = \{\mathcal{F}^2_n \circ \mathcal{F}^1_n \colon \mathsf{D}^1_n \mapsto \mathcal{R}^2_n\}_{n \in \mathbb{N}}$, where $\mathcal{F}^2_n \circ \mathcal{F}^1_n = \{(f_2, f_1) \in \mathcal{F}^2_n \times \mathcal{F}^1_n\}$, and $(f_2, f_1)(x) := f_2(f_1(x))$.*

Definition 2.3 (XOR of function families). *Let $\mathcal{F}^1 = \{\mathcal{F}^1_n \colon \mathsf{D}^1_n \mapsto \mathcal{R}^1_n\}_{n \in \mathbb{N}}$ and $\mathcal{F}^2 = \{\mathcal{F}^2_n \colon \mathsf{D}^2_n \mapsto \mathcal{R}^2_n\}_{n \in \mathbb{N}}$ be two ensembles of function families with $\mathcal{R}^1_n, \mathcal{R}^2_n \subseteq \{0,1\}^{\ell(n)}$ for every n. We define the* XOR *of $\mathcal{F}^1$ with $\mathcal{F}^2$ as $\mathcal{F}^2 \bigoplus \mathcal{F}^1 = \{\mathcal{F}^2_n \bigoplus \mathcal{F}^1_n \colon \mathsf{D}^1_n \cap \mathsf{D}^2_n \mapsto \{0,1\}^{\ell(n)}\}_{n \in \mathbb{N}}$, where $\mathcal{F}^2_n \bigoplus \mathcal{F}^1_n = \{(f_2, f_1) \in \mathcal{F}^2_n \times \mathcal{F}^1_n\}$, and $(f_2, f_1)(x) := f_2(x) \oplus f_1(x)$.*

2.2.2 Pairwise Independent Hashing

Definition 2.4 (pairwise independent families). *A function family $\mathcal{H} = \{h \colon \mathsf{D} \mapsto \mathcal{R}\}$ is* pairwise independent *(with respect to D and $\mathcal{R}$), if*

$$\Pr_{h \leftarrow \mathcal{H}}[h(x_1) = y_1 \wedge h(x_2) = y_2] = \frac{1}{|\mathcal{R}|^2},$$

for every distinct $x_1, x_2 \in \mathcal{D}$ and every $y_1, y_2 \in \mathcal{R}$.

For every $\ell \in \text{poly}$, the existence of efficient pairwise-independent families mapping strings of length n to strings of length $\ell(n)$, is well known ([3]). In this paper we use efficient pairwise-independent families mapping strings of length n to the set $[T(n)]_{\{0,1\}^n}$, where $T(n) \leq 2^n$. For the existence of such families, see Corollary 3.9.

2.2.3 Pseudorandom Functions

Definition 2.5 (pseudorandom functions)**.** *An efficient ensemble of function families $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ is a $(T(n), \varepsilon(n))$-PRF, if for every oracle-aided algorithm (distinguisher) D of running time $T(n)$ and large enough n, it holds that*

$$\left|\Pr_{f\leftarrow\mathcal{F}_n}[\mathsf{D}^f(1^n)=1] - \Pr_{\pi\leftarrow\Pi_n}[\mathsf{D}^\pi(1^n)=1]\right| \leq \varepsilon(n),$$

where the above probability is also over the random coins of D. If we limit D above to be non-adaptive (i.e., it has to write all his oracle calls before making the first call), then $\mathcal{F}$ is called $(T(n), \varepsilon(n))$-$\mathsf{non\text{-}adaptive\text{-}PRF}$.

The ensemble $\mathcal{F}$ is a t-PRF, if it is a $(t, 1/t)$-PRF according to the above definition, and it is simply a PRF, if it is a p-PRF for every $p \in \text{poly}$. The same conventions are also used for non-adaptive PRFs.

3 Our Construction

In this section we present the main contribution of this paper — a direct construction of (an adaptive) pseudorandom function family, from a non-adaptive one.

Theorem 3.1 (restatement of Theorem 1.1)**.** *Let $T(n)$ be a polynomial-time computable integer function, let $\mathcal{H}$ be an efficient pairwise independent function family mapping string of length n to $[T(n)]_{\{0,1\}^n}$ and let $\mathcal{F}$ be a $(p(n)\cdot T(n), \varepsilon(n))$-non-adaptive-PRF, where $p \in \text{poly}$ is determined by the computation time of T, $\mathcal{F}$ and H. Then $\mathcal{F}\circ\mathcal{H}$ is a $\left(s(n), \varepsilon(n) + 2\cdot\frac{s(n)^2}{T(n)-2s(n)^2}\right)$-PRF for every $s(n) < \sqrt{T(n)/2}$.*

Theorem 3.1 yields the following simpler statement.

Corollary 3.2. *Let T, p and $\mathcal{H}$ be as in Theorem 3.1. Assuming $\mathcal{F}$ is a $p(n)T(n)$-non-adaptive-PRF, then $\mathcal{F}\circ\mathcal{H}$ is a $\sqrt[3]{T(n)}/2$-PRF.*

Proof. Applying Theorem 3.1 with respect to $s(n) = \sqrt[3]{T(n)}/2$ and $\varepsilon(n) = 1/p(n)T(n)$, yields that $\mathcal{F}\circ\mathcal{H}$ is a $\left(s(n), \frac{1}{p(n)T(n)} + 2\cdot\frac{s(n)^2}{T(n)-2s(n)^2}\right)$-PRF. Since $\frac{1}{p(n)T(n)} < \frac{1}{2s(n)}$ and $\frac{2s(n)^2}{T(n)-2s(n)^2} \leq \frac{1}{3s(n)}$ (were for the latter we assume without loss of generality that $T(n) \geq 64$), it follows that $\mathcal{F}\circ\mathcal{H}$ is an $(s, 1/s)$-PRF. $\square$

To prove Theorem 3.1, we use the (non efficient) function family $\Pi\circ\mathcal{H}$, where $\Pi = \Pi_\ell$ (i.e., the ensemble of random functions from $\{0,1\}^n$ to $\{0,1\}^\ell$), and $\ell = \ell(n)$ is the output length of $\mathcal{F}$. We first show that $\mathcal{F}\circ\mathcal{H}$ is *computationally* indistinguishable from $\Pi\circ\mathcal{H}$, and complete the proof showing that $\Pi\circ\mathcal{H}$ is *statistically* indistinguishable from Π.

3.1 $\mathcal{F} \circ \mathcal{H}$ is Computationally Indistinguishable From $\Pi \circ \mathcal{H}$

Lemma 3.3. *Let T, $\mathcal{F}$ and H be as in Theorem 3.1. Then for every oracle-aided distinguisher D of running-time $T(n)$, there exists a* non-adaptive *oracle-aided distinguisher $\widehat{\mathsf{D}}$ of running-time $p(n) \cdot T(n)$, for some $p \in \mathrm{poly}$ that is determined by the computation time of T, $\mathcal{F}$ and H, with*

$$\left|\Pr_{g \leftarrow \mathcal{F}_n}[\widehat{\mathsf{D}}^g(1^n) = 1] - \Pr_{g \leftarrow \Pi_n}[\widehat{\mathsf{D}}^g(1^n) = 1]\right| = \left|\Pr_{g \leftarrow \mathcal{F}_n \circ \mathcal{H}_n}[\mathsf{D}^g(1^n) = 1] - \Pr_{g \leftarrow \Pi_n \circ \mathcal{H}_n}[\mathsf{D}^g(1^n) = 1]\right|$$

for every $n \in \mathbb{N}$.

In particular, the pseudorandomness of $\mathcal{F}$ yields that $\mathcal{F} \circ \mathcal{H}$ is computationally indistinguishable from $\Pi \circ \mathcal{H}$ by an adaptive distinguisher of running-time $T(n)$.

Proof. The distinguisher $\widehat{\mathsf{D}}$ is defined as follows:

Algorithm 3.4 ($\widehat{\mathsf{D}}$)**.**

Input: 1^n.

Oracle: *a function ϕ over $\{0,1\}^n$.*

1. *Compute $\phi(x)$ for all $x \in [T(n)]_{\{0,1\}^n}$.*
2. *Set $g = \phi \circ h$, where h is uniformly chosen in $\mathcal{H}_n$.*
3. *Emulate $\mathsf{D}^g(1^n)$: answer a query x to ϕ made by D with $g(x)$, using the information obtained in Step 1.*

Note that $\widehat{\mathsf{D}}$ only makes *non-adaptive* queries to ϕ, and that for large enough p in the statement of the lemma, it can be implemented to run in time $p(n)T(n)$. We conclude the proof observing that the emulation of D done in $\widehat{\mathsf{D}}^g$ is identical to $\mathsf{D}^{\mathcal{F}_n \circ \mathcal{H}_n}$, in case ϕ is uniformly drawn from $\mathcal{F}_n$, and to D^{Π_n}, in case ϕ is uniformly drawn from Π_n. □

3.2 $\Pi \circ \mathcal{H}$ is Statistically Indistinguishable From Π

Lemma 3.5. *Let D be an (unbounded) oracle-aided algorithm of running time $s(n)$. Assuming that $s(n) < \sqrt{T(n)/2}$, then*

$$\left|\Pr_{g \leftarrow \Pi_n \circ \mathcal{H}_n}[\mathsf{D}^g(1^n) = 1] - \Pr_{\pi \leftarrow \Pi_n}[\mathsf{D}^\pi(1^n) = 1]\right| \le 2 \cdot \frac{s(n)^2}{T - 2s(n)^2}.$$

Proof. We assume for simplicity that D is deterministic (the reduction to the randomized case is standard) and that on input 1^n, D makes exactly $s(n)$ valid (i.e., inside $\{0,1\}^n$) distinct queries. In the following we fix $n \in \mathbb{N}$ with $s(n) < \sqrt{T(n)/2}$, and omit n from notation when convenient.

For a function ϕ and integer $t \le s$, let $A_{\phi,t}$ denote the first t answers ϕ reply $\mathsf{D}^\phi(1^n)$, let $A_{\Pi \circ \mathcal{H},t} = (A_{g,t})_{g \leftarrow \Pi \circ \mathcal{H}}$ and let $A_{\Pi,t} = (A_{\pi,t})_{\pi \leftarrow \Pi}$. Since D is assumed to be deterministic, its output is determined by the answers to its oracle calls. Hence, the proof of the lemma immediately follows by the next claim (proof given below).

Claim 3.6. *For every $t \le s$ it holds that $\mathrm{SD}(A_{\Pi \circ \mathcal{H},t}, A_{\Pi,t}) \le \frac{t^2}{T - 2t^2}$.*

□

3.2.1 Proving Claim 3.6

For $t \in [s]$, let $\Omega(t) = (\{0,1\}^\ell)^t$ (recall that ℓ is the output length of $\mathcal{F}$). The proof of Claim 3.6 immediately follows by the next two claims:

Claim 3.7. *For every $t \leq s$, it holds that $A_{\Pi,t}$ is uniformly distributed over $\Omega(t)$.*

Claim 3.8. *For every $t \leq s$ and $\overline{a} \in \Omega(t)$, it holds that $A_{\Pi\circ\mathcal{H},t}(\overline{a}) \geq \left(1 - \frac{t^2}{T-t^2}\right) \cdot 2^{-t\ell}$.*

Before proving the above claims, let us first use them for proving Claim 3.6.

Proof of Claim 3.6. Claim 3.7 yields that $A_{\Pi,t}(\overline{a}) = 2^{-t\ell}$ for every $\overline{a} \in \Omega(t)$. Putting it together with Claim 3.8 yields that

$$\frac{A_{\Pi,t}(\overline{a})}{A_{\Pi\circ\mathcal{H},t}(\overline{a})} \leq \frac{1}{1 - \frac{t^2}{T-t^2}} = 1 + \frac{t^2}{T-2t^2}$$

for every $\overline{a} \in \Omega(t)$, yielding that $\mathrm{SD}(A_{\Pi\circ\mathcal{H},t}, A_{\Pi,t}) \leq \frac{t^2}{T-2t^2}$. $\square$

Proof of Claim 3.7. We prove by induction on t that $A_{\Pi,t}(\overline{a}) = 2^{-\ell t}$ for every $\overline{a} \in \Omega(t)$. The case $t = 1$ is immediate. Assuming for $t-1$, we note that since (by assumption) D does not make the same query twice, the answer of its t'th query is uniform in $\{0,1\}^\ell$ (regardless of what happened in the first $(t-1)$ queries). Therefore, $A_{\Pi,t}(\overline{a}) = A_{\Pi,t-1}(\overline{a}_{1\ldots t-1}) \cdot 2^{-\ell} = 2^{-\ell t}$ for every $\overline{a} \in \Omega(t)$. $\square$

Proof of Claim 3.8. Fix $\overline{a} \in \Omega(t)$, and let $\overline{q}$ be D's queries determined by $\overline{a}$. For $i \in [t]$ and $h \in \mathcal{H}$, let the indicator $\mathrm{Coll}(h,i)$ be one, if there exist $1 \leq j < j' \leq i$ with $h(q_j) = h(q_{j'})$. We prove the claim by showing that

$$\Pr_{\substack{h\leftarrow\mathcal{H}\\ \pi\leftarrow\Pi}}\left[\pi\circ h(\overline{q}_{1,\ldots,i}) = \overline{a}_{1,\ldots,i} \wedge \neg\mathrm{Coll}(h,i)\right] \geq \left(1 - \frac{i^2}{T-i^2}\right)\cdot 2^{-i\ell} \tag{1}$$

for every $i \in [t]$, where $\pi\circ h(\overline{q}_{1,\ldots,i}) = (\pi\circ h(\overline{q}_1), \ldots, \pi\circ h(\overline{q}_i))$. We prove Equation (1) by induction on i. The case $i = 1$ is immediate. Assuming for $i-1$, we write

$$\Pr\left[\pi\circ h(\overline{q}_{1,\ldots,i}) = \overline{a}_{1,\ldots,i} \wedge \neg\mathrm{Coll}(h,i)\right] = \Pr[\pi\circ h(\overline{q}_{1,\ldots,i-1}) = \overline{a}_{1,\ldots,i-1} \wedge \neg\mathrm{Coll}(h,i-1)\big] \cdot \alpha\beta, \tag{2}$$

for

$$\alpha = \Pr\left[\neg\mathrm{Coll}(h,i) \mid \neg\mathrm{Coll}(h,i-1) \wedge \pi\circ h(\overline{q}_{1,\ldots,i-1}) = \overline{a}_{1,\ldots,i-1}\right] \tag{3}$$

and

$$\beta = \Pr\left[\pi(h(q_i)) = a_i \mid \pi\circ h(\overline{q}_{1,\ldots,i-1}) = \overline{a}_{1,\ldots,i-1} \wedge \neg\mathrm{Coll}(h,i)\right], \tag{4}$$

where all probabilities are over uniformly chosen $(\pi, h) \in \Pi \times \mathcal{H}$. We show (see below) that $\alpha \geq 1 - \frac{i-1}{T-(i-1)^2}$ and $\beta = 2^{-\ell}$, and conclude that

$$\begin{aligned}
\Pr_{\substack{h\leftarrow\mathcal{H}\\ \pi\leftarrow\Pi}}\left[\pi\circ h(\overline{q}_{1,\ldots,i}) = \overline{a}_{1,\ldots,i} \wedge \neg\mathrm{Coll}(h,i)\right] &\geq \left(1 - \frac{(i-1)^2}{T-(i-1)^2}\right)2^{-(i-1)\ell}\cdot\left(1 - \frac{i-1}{T-(i-1)^2}\right)2^{-\ell}\\
&\geq \left(1 - \left(\frac{(i-1)^2}{T-(i-1)^2} + \frac{i-1}{T-(i-1)^2}\right)\right)2^{-i\ell}\\
&\geq \left(1 - \frac{i^2}{T-i^2}\right)2^{-i\ell},
\end{aligned}$$

as requested. We now conclude the proof by calculating α and β.

$\alpha \geq 1 - \frac{i-1}{T-(i-1)^2}$**:** Let $\mathcal{H}' = \{h \in \mathcal{H} \colon \neg\mathrm{Coll}(h, i-1)\}$. Note that the number of $\pi \in \Pi$ satisfying $\pi \circ h(\overline{q}_{1,\ldots,i-1}) = \overline{a}_{1,\ldots,i-1}$, is the same for every $h \in \mathcal{H}'$, and zero for $h \in \mathcal{H} \setminus \mathcal{H}'$. It follows that

$$\alpha = \Pr_{h \leftarrow \mathcal{H}'}[h(q_i) \notin \{h(q_j)\}_{j \in [i-1]}] \tag{5}$$

The pairwise independence of $\mathcal{H}$ yields that

$$\begin{aligned} \frac{i-1}{T} &= \Pr_{h \leftarrow \mathcal{H}}[h(q_i) \in \{h(q_j)\}_{j \in [i-1]}] \\ &\geq \Pr_{h \leftarrow \mathcal{H}}[\neg\mathrm{Coll}(h, i-1)] \cdot \Pr_{h \leftarrow \mathcal{H}'}[h(q_i) \in \{h(q_j)\}_{j \in [i-1]}], \end{aligned} \tag{6}$$

where a union bound yields that $\Pr_{h \leftarrow \mathcal{H}}[\mathrm{Coll}(h, i-1)] \leq \sum_{j \neq j' \in [i-1]} \Pr_{h \leftarrow \mathcal{H}}[h(q_j) = h(q_{j'})] \leq \frac{(i-1)^2}{T}$. We conclude that $\Pr_{h \leftarrow \mathcal{H}'}[h(q_i) \in \{h(q_j)\}_{j \in [i-1]}] \leq \frac{i-1}{T-(i-1)^2}$.

$\beta = 2^{-\ell}$**:** Let $\mathrm{Distinct}(i) \subseteq (\{0,1\}^n)^i$ be the subset of tuples with distinct elements (i.e., $\mathrm{Distinct}(i) = \{(b_1, \cdots, b_i) \in (\{0,1\}^n)^i \colon \forall 1 \leq j < j' \leq i \;\; b_j \neq b_{j'}\}$ and for $\overline{b} = (b_1, \ldots, b_i) \in \mathrm{Distinct}(i)$, let $\mathcal{S}_{\overline{b}} = \{(h, \pi) \in \mathcal{H} \times \Pi \colon \forall j \in [i] \;\; h(q_j) = b_j \wedge \forall j \in [i-1] \;\; \pi(b_j) = a_j\}$. Since the fraction of pairs inside $\mathcal{S}_{\overline{b}}$ with $\pi(b_i) = a$ is the same for every $a \in \{0,1\}^\ell$, it follows that

$$\Pr_{(h,\pi) \leftarrow \mathcal{S}_{\overline{b}}}[\pi(h(q_i)) = a_i] = 2^{-\ell} \tag{7}$$

for every non-empty $\mathcal{S}_{\overline{b}}$. Observing that $\beta = \Pr_{(h,\pi) \leftarrow \mathcal{S}}[\pi(h(q_i)) = a_i]$, for $\mathcal{S} = \bigcup_{\overline{b} \in \mathrm{Distinct}(i)} \mathcal{S}_{\overline{b}}$, and that the $\mathcal{S}_{\overline{b}}$'s are disjoint, Equation (7) yields that $\beta = 2^{-\ell}$.

□

3.3 Putting It Together

We are now finally ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let $s = s(n) \in \mathbb{N}$ be such that $s(n) < \sqrt{T(n)/2}$ for every $n \in \mathbb{N}$, and let D be an oracle-aided algorithm of running time $s(n)$. Lemma 3.3 yields that $|\Pr_{g \leftarrow \mathcal{F}_n \circ \mathcal{H}_n}[\mathsf{D}^g(1^n) = 1] - \Pr_{g \leftarrow \Pi_n \circ \mathcal{H}_n}[\mathsf{D}^g(1^n) = 1]| \leq \varepsilon(n)$ for large enough n, where Lemma 3.5 yields that $|\Pr_{g \leftarrow \Pi_n \circ \mathcal{H}_n}[\mathsf{D}^g(1^n) = 1] - \Pr_{\pi \leftarrow \Pi_n}[\mathsf{D}^\pi(1^n) = 1]| \leq 2 \cdot \left(\frac{s(n)^2}{T(n) - 2s(n)^2}\right)$ for every $n \in \mathbb{N}$. Hence, the triangle inequality yields that $|\Pr_{g \leftarrow \mathcal{F}_n \circ \mathcal{H}_n}[\mathsf{D}^g(1^n) = 1] - \Pr_{\pi \leftarrow \Pi_n}[\mathsf{D}^\pi(1^n) = 1]| \leq \varepsilon(n) + 2 \cdot \left(\frac{s(n)^2}{T(n) - 2s(n)^2}\right)$ for large enough n, as requested. □

3.4 Handling Unknown Security

Corollary 3.2 is useful when the security of the underlying non-adaptive PRF (i.e., T) is efficiently computable. In this section we show how to handle the general case, where nothing is assumed about the computability of T. Our construction, stated below, essentially tries "all" possible values for T, and combines them into a single family.

Given a length-preserving function family $\mathcal{H}$ over $\{0,1\}^n$ and an integer $i \leq \log n$, we let $\mathcal{H}^i$ be the function family $\mathcal{H}^i = \{\widehat{h} \colon h \in \mathcal{H}\}$, where $\widehat{h}(x)$ is the $h(x)_{1,\ldots,\log(m_n(i))}$'th element inside $\{0,1\}^n$, and $m_n(i)$ is the largest power of two that is smaller than n^{2^i}.

We prove the following corollary.

Corollary 3.9 (restatement of Corollary 1.2)**.** *Let $\mathcal{F}$ be a $T(n)$-non-adaptive PRF, let $\mathcal{H}$ be an efficient length-preserving pairwise-independent function family and let $k(n) \leq \mathrm{poly}(n)$ be polynomial-time computable integer function. Let G be the function-family ensemble $\{G_n\}_{n\in\mathbb{N}}$, where $G_n = \bigoplus_{i\in[k(n)]} \mathcal{F}_n \circ \mathcal{H}_n^i$.*

Then G is a $\sqrt[3]{m_n(i)}/2$-adaptive PRF, for every polynomial-time computable integer function $i(n) \leq k(n)$ with $m_n(i) \leq T(n)/q(n)$, where $q \in \mathrm{poly}$ is universal.

Proof. It is easy to see that G is efficient. We let $q(n) = q'(n)p(n)$, where p is as in the statement of Corollary 3.2, and $q' \in \mathrm{poly}$ to be determined later. Let $i(n) \leq k(n)$ be a polynomial-time computable integer function with $m_n(i) \leq T(n)/q(n)$ and let $\mathcal{H}^i = \{\mathcal{H}_n^{i(n)}\}_{n\in\mathbb{N}}$. Corollary 3.2 yields that $\mathcal{F}\circ\mathcal{H}^i$ is a $\sqrt[3]{q'(n)m_n(i)}/2$-adaptive PRF. Now assume towards a contradiction that there exists an oracle-aided distinguisher D that runs in time $T'(n) = \sqrt[3]{m_n(i)}/2$ and

$$|\Pr_{g\leftarrow G_n}[\mathsf{D}^g(1^n)=1] - \Pr_{g\leftarrow\Pi_n}[\mathsf{D}^g(1^n)=1]| > \frac{1}{T'(n)} \tag{8}$$

for infinitely many n's. We use the following distinguisher for breaking the pseudorandomness of $\mathcal{F}\circ\mathcal{H}^i$:

Algorithm 3.10 ($\widehat{\mathsf{D}}$)**.**

Input: 1^n.

Oracle: *a function ϕ over $\{0,1\}^n$.*

1. *For every j in $[k]\setminus\{i\}$, choose uniformly at random $g^j \in \mathcal{F}_n\circ\mathcal{H}_n^j$.*
2. *Set $g := g^1 \oplus \ldots \oplus g^{i-1} \oplus \phi \oplus g^{i+1} \oplus \ldots \oplus g^k$.*
3. *Emulate $\mathsf{D}^g(1^n)$.*

. . . .

Note that $\widehat{\mathsf{D}}$ can implemented to run in time $k(n)\cdot r(n)\cdot T'(n)$ for some $r \in \mathrm{poly}$, which is smaller than $\sqrt[3]{q'(n)m_n(i)}/2$ for large enough q'. Also note that in case ϕ is uniformly distributed over Π_n, then g (selected by $\widehat{\mathsf{D}}^\phi(1^n)$) is uniformly distributed in Π_n, where in case ϕ is uniformly distributed in $\mathcal{F}_n\circ\mathcal{H}_n^{i(n)}$, then g is uniformly distributed in G_n. It follows that

$$\left|\Pr_{g\leftarrow(\mathcal{F}\circ\mathcal{H}^i)_n}[\widehat{\mathsf{D}}^g(1^n)=1] - \Pr_{\pi\leftarrow\Pi_n}[\widehat{\mathsf{D}}^\pi(1^n)=1]\right| = |\Pr_{g\leftarrow G_n}[\mathsf{D}^g(1^n)=1] - \Pr_{g\leftarrow\Pi_n}[\mathsf{D}^g(1^n)=1]| \tag{9}$$

for every $n\in\mathbb{N}$. In particular, Equation (8) yields that

$$\left|\Pr_{g\leftarrow(\mathcal{F}\circ\mathcal{H}^i)_n}[\widehat{\mathsf{D}}^g(1^n)=1] - \Pr_{\pi\leftarrow\Pi_n}[\widehat{\mathsf{D}}^\pi(1^n)=1]\right| > \frac{2}{\sqrt[3]{m_n(i)}} > \frac{2}{\sqrt[3]{q'(n)m_n(i)}}$$

for infinitely many n's, in contradiction to the pseudorandomness of $\mathcal{F}\circ\mathcal{H}^i$ we proved above. □

3.4.1 Super-Polynomial Security

Let $\mathcal{F}$ be a non-adaptive PRF (i.e., $\mathcal{F}$ is indistinguishable from random by any non-adaptive polynomial-time distinguisher). Applying Corollary 3.9 with respect to the above $\mathcal{F}$, an efficient length-preserving pairwise-independent function family $\mathcal{H}$ and a polynomial-time computable $k(n) = \omega(1)$ (e.g., $k(n) = \log^*(n)$), yields that G is $\sqrt[3]{m_n(c)}/2$-adaptive PRF any constant $c \in \mathbb{N}$. It follows that G is p-adaptive PRF for every $p \in \text{poly}$, and thus a PRF.

Acknowledgment

We are very grateful to Omer Reingold for very useful discussions, and for challenging the second author with this research question a long while ago.

References

[1] M. Bellare and S. Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In *Advances in Cryptology – CRYPTO '89*, pages 194–211, 1989.

[2] M. Blum, W. S. Evans, P. Gemmell, S. Kannan, and M. Naor. Checking the correctness of memories. *Algorithmica*, 12(2/3):225–244, 1994.

[3] L. J. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, pages 143–154, 1979.

[4] C. Cho, C.-K. Lee, and R. Ostrovsky. Equivalence of uniform key agreement and composition insecurity. In *Advances in Cryptology – CRYPTO 2010*, pages 447–464, 2010.

[5] B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Transactions on Information Theory*, 46(3):893–910, 2000.

[6] O. Goldreich. Towards a theory of software protection. In *Advances in Cryptology – CRYPTO '86*, pages 426–439, 1986.

[7] O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.

[8] O. Goldreich, S. Goldwasser, and S. Micali. On the cryptographic applications of random functions. pages 276–288, 1984.

[9] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, pages 792–807, 1986.

[10] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, pages 1364–1396, 1999.

[11] M. Luby. *Pseudorandomness and cryptographic applications*. Princeton computer science notes. Princeton University Press, 1996. ISBN 978-0-691-02546-9.

[12] M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*.

[13] U. M. Maurer and K. Pietrzak. Composition of random systems: When two weak make one strong. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, pages 410–427, 2004.

[14] S. Myers. Black-box composition does not imply adaptive security. In *Advances in Cryptology – EUROCRYPT 2004*, pages 189–206, 2004.

[15] M. Naor and O. Reingold. Synthesizers and their application to the parallel construction of psuedo-random functions. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 170–181, 1995.

[16] R. Ostrovsky. An efficient software protection scheme. In *Advances in Cryptology – CRYPTO '89*, 1989.

[17] K. Pietrzak. Composition does not imply adaptive security. In *Advances in Cryptology – CRYPTO 2005*, pages 55–65, 2005.

[18] K. Pietrzak. Composition implies adaptive security in minicrypt. In *Advances in Cryptology – EUROCRYPT 2006*, pages 328–338, 2006.